QMARK 201.2

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s)	:	Shepherd, et al.
Serial No.	:	10/791,019
Filed	:	March 2, 2004
For	:	SECURE BROWSER
Examiner	:	Haoshian Shih
Art Unit	:	2173
Customer No.	:	10037

February 25, 2008

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

APPEAL BRIEF UNDER 37 C.F.R. § 41.37

In response to the Final Office Action dated November 26, 2007, and the Notice of Appeal dated April 28, 2008, applicants provide herewith their Appeal Brief.

(i) Real party in interest.

The real party in interest is Question Mark Computing Limited. An assignment is recorded at Reel 015054, frame 0472.

(ii) Related appeals and interferences.

None.

(iii) Status of claims.

Claims 1-21 are in the application.

Claims 1-21 are rejected.

The rejection of claims 1-21 is appealed,.

(iv) Status of amendments.

The amendment after final rejection is entered on appeal. See Advisory Action dated April 1, 2008.

(v) Summary of claimed subject matter.

Claim 1 provides a secure user interface method, for interacting with a user through a browser (IE in Fig. 2), comprising:

controlling the browser (Workstation in Fig. 1) to request a document ("secure content" in Figs. 1 and 2), from a cooperative server (Server in Fig. 1), the browser providing data export support functionality;

receiving data with the browser in response to the request ("Server responds" in Fig. 1);

automatically determining, based on a received data encoding type, whether a secure browser ("secure browser" in Fig. 1) or a normal browser is to be employed (QMSB MIME type), the secure browser having a set of functionality restricted with respect to the normal browser, to enhance security of a received document against data export (page 9, line 25-page 10, line 2);

receiving the secure content for presentation in the secure browser; and

communicating an input from the user, through the secure browser, to a cooperative server (Page 4, line 11-page 5, line 19).

Claim 9 provides a secure user interface method, for interacting with a user through a browser (IE in Fig. 2), the browser providing a set of navigational functionality, comprising:

requesting a document ("secure content" in Figs. 1 and 2) from a cooperative server (Server in Fig. 1);

receiving data in response to the request ("Server returns web page" in Fig. 1);

automatically determining, based on a received data type encoding (MIME type of QMSB

in Fig. 1), whether a secure browser ("Secure Browser" in Figs. 1 and 2) is required to be employed by a content provider (Server in Fig. 1, "Authenticate" in Fig. 2) or whether an insecure browser is to be employed ("Original Browser" in Fig. 2), the secure browser restricting interaction of the user with tasks other than those permitted by the secure browser which are permitted by the insecure browser (page 9, line 25-page 10, line 2);

invoking the secure browser ("QSB called" in Fig. 2);

receiving the secure content for presentation in the secure browser ("Server delivers secure content" in Fig. 1, "deliver secure content" in Fig. 2)); and

communicating an input from the user, through the secure browser, to a cooperative server (Page 4, line 11-page 5, line 19).

(vi) Grounds of rejection to be reviewed on appeal.

Claims 1-4 and 6-20 are rejected as being anticipated under 35 U.S.C. § 102(e) over Winneg et al., US 7,069,586.

Claims 5 and 21 are rejected as being obvious under 35 U.S.C. § 103(a) over Winneg et al., US 7,069,586 in view of Chang et al., US 2002/0097416.

(vii) Argument.

REJECTION OF CLAIMS 1-4 AND 6-20 UNDER 35 U.S.C. § 102(E).

Winneg et al. is respectfully distinguished in that a "professor" and a "student" can access the very same document (having the same type encoding) and be afforded different privileges based on their login; in accordance with the presently claimed invention, it is the *data* which determines the browser type, not the *user*, which in turn defines the set of privileges available through the selected browser.

In formulating the rejection, the Examiner cites various portions of Winneg, which it is respectfully submitted do not teach or suggest at least the foregoing limitation. For example, the Examiner cites Col. 4, lines 3-5 for the proposition that Winneg teaches "automatically determining, based on a type encoding of the received data, whether a secure browser or a normal browser is to be employed". However, at this passage, Winneg states: "The application being securely executed may be of any of a variety of types of applications, for example, a browser application or an application for receiving answers to questions of an examination (i.e., an exam taking application)." Thus, while Winneg appears to disclose a secure browser mode, it fails to disclose that a normal or insecure mode is also selectively available, in dependence on a type of encoding or by a content provider, having a different level of functionality.

The secure mode of Winneg appears to be initiated based on a boot sequence, operating system limitation or user login. Col. 6, lines 35-67. Col. 9, lines 45-47, 50-55 and Col. 10, lines 10-13 indicate that a **user input** (and not a type encoding) determines which application to initiate. ("For example, FIG. 7 illustrates a GUI that may be displayed to a user to determine which application to initiate for the exam." "After the user has entered the class name and the professor in their respective fields and clicked on the OK button, the exam-taking application

may use this information to determine a first application to be executed so that the student may take the exam (i.e., provide responses to one or more questions) and to determine the content (e.g., the questions of the exam or material to assist the user in taking the exam), if any, to be displayed by the first application." "Else, after hitting the 'OK' button of the GUI, next, in Act 122, secure execution of the exam-taking application may be initiated.").

Winneg et al. appears to provide a system in which a local software application controls the client computer independent of a type encoding of the received data. For example, Col. 6, lines 35-48 describe a system which defaults to a "secure" mode, and is machine status dependent, not received data dependent. Indeed, the authorization to access or delete an exam is provided within the "secure" mode, and thus these functions are all provided within a single "browser" or its analog. Therefore, the decisions 114, 116 do not serve to switch "browsers". Col. 8, lines 48- Col. 9, line 44. Throughout the entire exam process, the machine is locked in a "secure" mode, maintaining this mode apparently independent of received data.

CLAIMS 1-8

The examiner interprets the phrase "type encoding" in claim 1 to encompass a login classification of the user. This, however, is an erroneous interpretation of the claim. The complete claim phrase of claim 1 is "…based on a type encoding of the received data…" Therefore, this language does not relate to a type of **USER**, but rather a type of **DATA**. Likewise, the result of this type encoding in accordance with the claims is the selection of a secure browser or a normal browser with respectively different level of functionality, and not a set of privileges of the user within a singular browser type.

It is especially noted that, in accordance with claim 1, the decision of whether to employ a secure browser or a normal browser is automatically determined based on a type of encoding of the received data. Therefore, it is not the server, but the respective client, which automatically determines which browser to employ, and that this determination is not automatically made based on a type encoding of the data received by the browser.

Col. 9, lines 59-67 provide that it is the information entered in the fields of Fig. 7 that are used to determine if content is to be displayed by "the first application" (e.g., MS Word). Fig. 7 shows a login screen, in which a user enters class name, professor and exam date. This does not correspond to the document requested by the browser from the cooperative server, and received by the browser in response to the request, as provided by claim 1.

Therefore, it is seen that Winneg et al. employ a presumption that, so long as the exam-taking application is engaged, the machine must be in the "secure" mode, and do not employ encoding of requested data received from the server to automatically control whether a secure browser or insecure browser is employed. . This differs from the present invention in accordance with claim 1, which permits, for example, the server to dynamically control the browser based on data encoding.

The dependent claims 2-4 and 6-8 are believed to be distinguished at least on the same basis.

CLAIMS 9-20

The examiner interprets the phrase "type encoding" in claim 9 to encompass a login classification of the user. This, however, is an erroneous interpretation of the claim. The complete claim phrase of claim 9 is "automatically determining, based on a received data type

encoding, whether a secure browser is required to be employed by a content provider or whether an insecure browser is to be employed, the secure browser restricting interaction of the user with tasks other than those permitted by the secure browser which are permitted by the insecure browser".

Therefore, this language does not relate to a type of **USER**, but rather a type of **DATA**. Likewise, the result of this type encoding in accordance with the claims is the selection of a secure browser or a normal browser with respectively different level of functionality, and not a set of privileges of the user within a singular browser type.

It is especially noted that, in accordance with claim 9, the decision of whether to employ a secure browser or a normal browser is automatically determined based on a type of encoding of the received data. Therefore, it is not the server, but the respective client, which automatically determines which browser to employ, and that this determination is not automatically made based on a type encoding of the data received by the browser.

Therefore, it is seen that Winneg et al. employ a presumption that, so long as the exam-taking application is engaged, the machine must be in the "secure" mode, and do not employ encoding of requested data received from the server to automatically control whether a secure browser or insecure browser is employed. . This differs from the present invention in accordance with claim 9, which permits, for example, the server to dynamically control the browser based on data encoding.

Claim 9 is particularly distinguished in that Winneg et al. employ only a single browser type, and not both a separately defined secure browser and an insecure browser, the use of which is determined automatically by a content provider. As discussed above, the decision by Winneg et al. of whether to employ a security or not is made in dependence on a login status, and

therefore, Winneg et al. do not teach or suggest at least "automatically determining whether a secure browser is required to be employed by a content provider or whether an insecure browser is to be employed, the secure browser restricting interaction of the user with tasks other than those permitted by the secure browser which are permitted by the insecure browser," as provided in claim 9.

The dependent claims 10-20 are believed to be distinguished at least on the same basis.

REJECTION OF CLAIMS 5 AND 21 UNDER 35 U.S.C. § 103(A)

In a typical computer system, it is not the browser which converts "text information" to "graphic objects"; rather, there is a display driver system separate from the application, which receives the source information to be displayed, and then formats it for presentation. Therefore, similar to the architecture of Chang et al., the rasterizer services a plurality of applications, and is separate therefrom

CLAIM 5.

Claim 5 is distinguished at least on the same basis as claim 1.

In addition, it is noted that claim 5 provides a different configuration from Chang et al., in which the rasterizer (conversion from text information to a graphic object) is part of the secure browser process, i.e., "the secure browser renders text information as graphic objects." Therefore, Winneg et al., US 7,069,586 and Chang et al., US 2002/0097416 are believed distinguished. In particular, it is not believed that Chang et al. disclose an application-level rasterizer, but rather a rasterizer that services all applications on a device, and therefore is distinguished by claim 5 which requires that the secure browser, an application, perform the conversion.

CLAIM 20.

Claim 20 is distinguished at least on the same basis as claim 9.

In addition, it is noted that claim 5 provides a different configuration from Chang et al., in which the rasterizer (conversion from text information to a graphic object) is part of the secure browser process, i.e., "the secure browser renders text information as graphic objects."

Therefore, Winneg et al., US 7,069,586 and Chang et al., US 2002/0097416 are believed distinguished. In particular, it is not believed that Chang et al. disclose an application-level rasterizer, but rather a rasterizer that services all applications on a device, and therefore is distinguished by claim 5 which requires that the secure browser, an application, perform the conversion.

Therefore, it is respectfully requested that the rejection of the claims be reconsidered and withdrawn.

Respectfully submitted,

/Steven M. Hoffberg/
Steven M. Hoffberg
Reg. No. 33,511

MILDE & HOFFBERG, LLP
10 Bank Street-Suite 460
White Plains, NY 10606
(914) 949-3100

(viii) Claims appendix.

1. A secure user interface method, for interacting with a user through a browser, comprising:

controlling the browser to request a document from a cooperative server, the browser providing data export support functionality;

receiving data with the browser in response to the request;

automatically determining, based on a received data encoding type, whether a secure browser or a normal browser is to be employed, the secure browser having a set of functionality restricted with respect to the normal browser, to enhance security of a received document against data export;

receiving the secure content for presentation in the secure browser; and

communicating an input from the user, through the secure browser, to a cooperative server.

2. The user interface method according to claim 1, further comprising the step of limiting access of a user, with the secure browser, to documents outside of a specified set.

3. The user interface method according to claim 1, further comprising the step of authenticating the secure browser, to assure that the secure browser having the restricted set of functionality is available for presentation of secure content.

4. The user interface method according to claim 1, wherein the secure browser lacks one or more of the following functions: print, save, cache, cut and copy.

5. The user interface method according to claim 1, wherein the secure browser renders text information as graphic objects.

6. The user interface method according to claim 1, wherein the secure browser restricts termination of execution of the secure browser.

7. The user interface method according to claim 3, wherein the secure browser restricts termination of execution of the secure browser.

8. A computer readable media storing a program for a general purpose computer for performing the method of claim 3

9. A secure user interface method, for interacting with a user through a browser, the browser providing a set of navigational functionality, comprising:

requesting a document from a cooperative server;

receiving data in response to the request;

automatically determining, based on a received data type encoding, whether a secure browser is required to be employed by a content provider or whether an insecure

browser is to be employed, the secure browser restricting interaction of the user with tasks other than those permitted by the secure browser which are permitted by the insecure browser;

invoking the secure browser;

receiving the secure content for presentation in the secure browser; and

communicating an input from the user, through the secure browser, to a cooperative server.

10. The method according to claim 9, wherein the secure browser provides restricted navigational functionality with respect to the navigational functionality of the insecure browser alone.

11. The user interface method according to claim 9, further comprising the step of limiting access of a user, with the secure browser, to access of documents within a specified set.

12. The user interface method according to claim 9, further comprising the step of authenticating the secure browser at a remote server prior to presenting the secure content to ensure that the content will only be delivered in the secure browser.

13. The user interface method according to claim 9, wherein the secure browser prevents use of the following functions: save, copy, and navigate to unrestricted

documents.

14. The user interface method according to claim 9, wherein the secure browser restricts termination of execution of the secure browser.

15. The user interface method according to claim 9, wherein the secure browser is initiated based on a type encoding of the received data.

16. The user interface method according to claim 9, wherein the secure browser is initiated based on a code associated with the secure content.

17. The user interface method according to claim 9, wherein the secure browser is granted principal application level control over graphic user interface inputs from a user.

18. The user interface method according to claim 9, wherein the secure browser is granted exclusive control over graphic user interface functionality when invoked.

19. The user interface method according to claim 9, further comprising the step of authenticating the server by the secure browser prior to presenting the secure content.

20. A computer readable media storing a program for a general purpose computer for performing the method of claim 9.

21. The user interface method according to claim 9, wherein the secure browser renders text information as graphic objects.

(ix) Evidence appendix.

Not Applicable.

(x) Related proceedings appendix.

Not Applicable.